Worksheet 3 Programming paradigms Answers

Task 1

1. Procedural languages were the first type of high level programming language to be developed in the 1950s-60s.

   Do some research to find out when and why each of the languages shown in the table on the next two pages was developed, and fill in the table.

   Tip: Look up on the Internet “History of programming languages”. There are some good PowerPoint presentations on the subject.

2. Can you find the name of a language that is designed to support concurrency (many processes running in parallel)?

   You can give students a hint if they cannot find the answer – Charles Babbage’s colleague, the first computer programmer, was called Ada Lovelace. **Ada** is a language that supports concurrency.

3. What languages are used for web programming?

   HTML (a scripting language), PHP, JavaScript, Perl, Python, ColdFusion, Ruby, ASP.net and more.

Language	Type	Developed/ released	Main usage	Brief description
ALGOL	Procedural/ imperative	1960	Designed to solve mathematical and engineering problems.	Not good at common tasks in business applications – for example validating input data, creating and maintaining files of structured data (records) and producing nicely formatted reports.
FORTRAN	Procedural/ imperative	1957	Numeric, scientific and engineering applications.	See above.
COBOL	Procedural/ imperative	Early 1960s	Business applications.	Payroll was one of the first applications for which computers were used, and **COBOL** (**CO**mmon **B**usiness **O**riented **L**anguage) was developed specifically for this type of application.
BASIC	Procedural/ imperative	1960s	It was developed as a simple language for students to learn to program.	**BASIC** (**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) has changed a lot since its early days, with several new variations which are no longer a "beginner's language".
C	Procedural/ imperative	1960s	To write operating systems and compilers.	It combined the capabilities of assembly language with features of a high-level language.

C++	Object oriented/ imperative	1990s	Systems programming and embedded systems.	**C++** is C with additional features including a graphical user interface. It provides facilities for low-level memory manipulation.
SQL	Declarative	1970s	To manage data in a database.	It has statements to insert, query, update and delete database schema and data.
Java	Object oriented	1990s	Used to teach students to program. Java applets which are stored on a website and run on client computer.	Java applications are usually compiled to bytecode that can run on any Java Virtual Machine (JVM). Java is currently one of the most popular programming languages being used. It has about 10 million users.
Haskell	Functional	2000s	Wide range of applications from aerospace and defence to finance and web startups.	Good for all sorts of data analysis problems.

Task 2

4. Fill in the table to compare the two programming paradigms

Paradigm: Procedural	Paradigm: Declarative
The statements generally have to be written in a particular sequence	The facts and rules can be written in any sequence
The programmer codes an algorithm which solves the problem	The programmer states the facts and rules associated with the problem
The programmer defines the steps that need to be taken to solve the problem	The programmer states the query and lets the program figure out the solution
The program always takes the route determined by the programmer	The program will try one route through the facts and rules and if that does not produce an answer, it will __backtrack____ and try another route until the problem is solved or __no more routes remain to be explored_____
Suitable for a wide variety of problems	Suitable for expert systems, medical diagnosis, natural language processing
Examples of this paradigm are ___Pascal, Python, VB__	Examples of this paradigm are ___SQL, Prolog_____

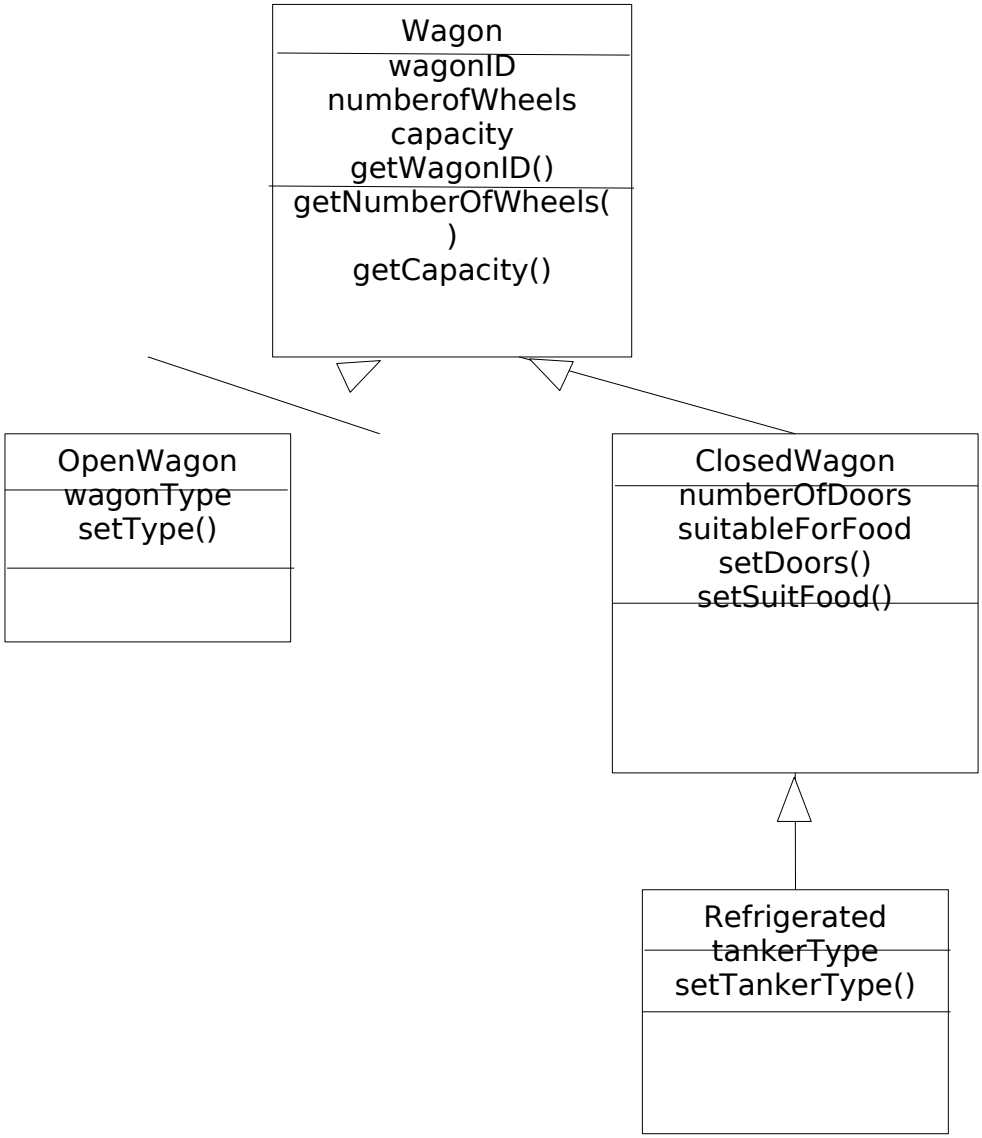
Prolog can be downloaded free, and there are several good Prolog tutorials on the Web. e.g.

http://www.doc.gold.ac.uk/~mas02gw/prolog_tutorial/prologpages/index.html#menu

Task 3

5. An example of an inheritance diagram is shown below.

(Note: a wagon is a “car” or “truck” used for carrying freight by rail.)



(a) What programming paradigm uses classes and objects?

The object-oriented paradigm

(b) Using the diagram above, explain the terms class, method, attribute, inheritance and encapsulation

Class: a template or blueprint for an object, which defines the attributes and methods of objects in that class.

Method: this is a procedure / function which can be carried out by any object in that class. It may or may not have parameters. Setter methods set the value of attributes, and getter methods return the value of an attribute.

Methods are generally declared **public** so that they can be used in any program using the class.

In the above inheritance diagram, getWagonID() is an example of a setter method.

getcapacity() is an example of a getter object.

Attribute – this is a property of an object in the class. Objects in the class can set the data value of an attribute, or examine its value, via setter and getter methods

Attributes are generally declared **private**, which means that they cannot be changed directly. They can only be changed through a public method.

Examples of attributes in the inheritance diagram include numberOfWheels, capacity, tankerType.

Inheritance: Classes can inherit data and behaviours (attributes and methods) from a parent class. A "child" class is referred to as a **subclass**, and a "parent" class as a **superclass**. Inheritance is a very useful concept because it means that many different classes can inherit from a superclass and therefore their attributes and methods do not have to be defined again, cutting out duplication of effort.

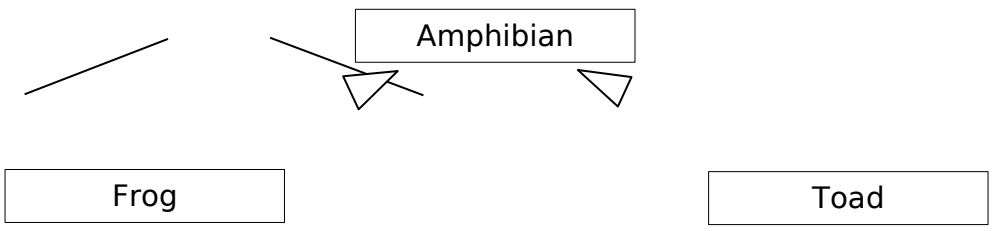
OpenWagon and ClosedWagon are subclasses of Wagon (the superclass). Refrigerated is a subclass of ClosedWagon and inherits all the methods and attributes of the superclasses Wagon and ClosedWagon.

The subclasses may additionally have methods and attributes of their own. e.g. In the above diagram, OpenWagon has an additional attribute type which could be, say, flatbed or enclosed.

Encapsulation: The object encapsulates both the state (values of the attributes) and the behaviours (methods) of an object. The attributes and methods of one object cannot affect the way in which another object functions.

Related to encapsulation is the concept of information hiding, whereby details of an object's attributes cannot be directly changed – they can only be changed through a public method.

6. Shown below is an inheritance diagram.



Amphibian has an attribute `position` and a method `jump`.

When an object in the `Frog` class calls the method `jump`, position is incremented by 3, but when in the `toad` class calls the method `jump`, position is incremented by 1.

Explain how this can be implemented.

It can be implemented through polymorphism – a method of the name `jump` is defined in the superclass amphibian may be defined as incrementing `position` by 1. This method will be inherited by the `toad` class. In the `frog` class, a new method called `jump` is defined which increments `position` by 3.

The change in `position` will therefore depend on what type of object calls the method `jump`,

What is the name given to a programming language's ability to process objects differently depending on their class?

Polymorphism